# Collaborative Training Between Region Proposal Localization and Classification for Domain Adaptive Object Detection

Ganlong Zhao, Guanbin Li(✉), Ruijia Xu, and Liang Lin

School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
{zhaoglong,xurj3}@mail2.sysu.edu.cn, liguanbin@mail.sysu.edu.cn, linliang@ieee.org

**Abstract.** Object detectors are usually trained with large amount of labeled data, which is expensive and labor-intensive. Pre-trained detectors applied to unlabeled dataset always suffer from the difference of dataset distribution, also called domain shift. Domain adaptation for object detection tries to adapt the detector from labeled datasets to unlabeled ones for better performance. In this paper, we are the first to reveal that the region proposal network (RPN) and region proposal classifier (RPC) in the endemic two-stage detectors (e.g., Faster RCNN) demonstrate significantly different transferability when facing large domain gap. The region classifier shows preferable performance but is limited without RPN's high-quality proposals while simple alignment in the backbone network is not effective enough for RPN adaptation. We delve into the consistency and the difference of RPN and RPC, treat them individually and leverage high-confidence output of one as mutual guidance to train the other. Moreover, the samples with low-confidence are used for discrepancy calculation between RPN and RPC and minimax optimization. Extensive experimental results on various scenarios have demonstrated the effectiveness of our proposed method in both domain-adaptive region proposal generation and object detection. Code is available at https://github.com/GanlongZhao/CST_DA_detection.

**Keywords:** Domain adaptation · Object detection · Transfer learning

## 1 Introduction

Benefiting from massively well-labeled data, deep convolutional neural networks have recently shown unparalleled advantages in various visual tasks, e.g., image recognition and object detection. Unfortunately, such data is usually prohibitive in many real-world scenarios. The problem becomes extremely serious for object detection, since it requires more precise object-level annotations. A common solution for this problem is to transfer the pretrained model from label-rich domain(i.e. source domain) to the other(i.e. target domain), but this often suffers from performance degradation due to domain gap.


A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12363, pp. 86–102, 2020.
https://doi.org/10.1007/978-3-030-58523-5_6

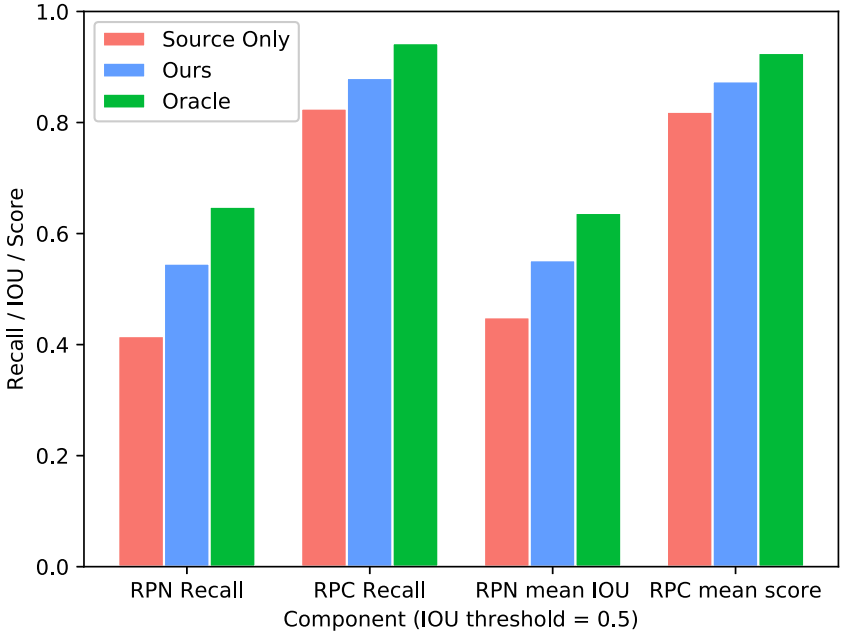



**Fig. 1.** Comparison of domain adaptation performance between "Source Only" model, "Ours" and the "Oracle" model w.r.t "Recall", "Average IOU" and "Mean Score". Threshold of IOU is set to 0.5 for "Recall" calculation. Experiment is conducted on adapting models trained on Sim10k [14] to Cityscapes [4].

Various kinds of methods have been proposed to overcome the domain gap. Most of them are based on adversarial training, and can be separated into two categories: feature-level and pixel-level domain adaptation. Feature-level domain adaptation tries to align the feature distributions from the two domains by adversarial training, while pixel-level domain adaptation uses GANs [10] to generate target-like images from source domain with labels unchanged.

There have been some research works focusing on domain adaptive object detection with feature-level or pixel-level adaptation techniques. Chen *et al.* [3] attempt to reduce domain discrepancy in both image level and instance level. Saito *et al.* [25] leverage weak global feature alignment and strong local alignment to mitigate the performance degradation caused by distinct scene layouts and different combinations of objects. Zhu *et al.* [34] propose to bridge the domain gap through effective region mining and region-based domain alignment. Noted that most of the previous research works on domain adaptive object detection focus on bridging the whole-image representations and thus perform alignment in the backbone branch of a detector. Though Zhu *et al.* [34] propose a selective adaptation framework based on region patches generated by the RPN branch, the loss gradient of the region classifier is just propagated to the backbone without adapting the RPN itself. Saito *et al.* [25] conduct feature alignment only on the

lower and final layer of the backbone. However, different from neural network based classification models, most of endemic two-stage object detectors are far more complex. It is far from sufficient to align and adapt the global features of the backbone network, which ignores the transferability of the RPN module.

RPN transferability is paramount for the adaptation of two-stage detectors, while adapting on the entire image with backbone-only alignment is not an effective solution. A two-stage object detector can be separated into three modules: backbone network, RPN and region proposal classifier (abbr. "RPC"). With large domain gap and complicated scene, we empirically discover that RPN and RPC show different transferability, i.e., RPC usually performs better than RPN. We adopt the "Source Only" domain adaptation model to investigate the transferability difference of RPN and RPC. Specifically, we directly apply the model trained on Sim10k [14] to test the performance on Cityscapes [4], take 0.5 as the IOU threshold and compute the recall of RPN and RPC respectively[1]. As shown in Fig. 1, it is obvious that RPC performs better than RPN before and after adaptation, and more importantly, has much less degradation between oracle(green bar) and the source-only model(red bar), which implies that RPN suffers much severer than RPC from domain gap. However, the performance of RPC is also limited if RPN fails to provide high-quality region proposals. RPN has therefore become the bottleneck. Noted that RPC here is not doomed to be better than RPN, because it considers the classification recall of the detected region proposals (even if the RPN detection is accurate, the accuracy of the proposal classification is still uncertain).

On the other hand, it is noteworthy that, in some kinds of two-stage object detectors (e.g., Faster RCNN), there is no gradient flow between RPN and RPC. A natural idea is to take them as two individual branches from backbone. If we consider RPN as a kind of foreground/background classifier, it can be regarded as a coarse and fast RPC classifying each anchor across the feature map. Similarly, if we sum up the output of RPC stream to background and foreground scores, it performs just like a fine-grained and selective RPN. Based on the above discussion, we propose a novel domain adaptation method on Faster RCNN using collaborative training between RPN and RPC. It can also be easily generalized to other two-stage detectors. Specifically, we first apply collaborative self-training between RPN and RPC, which leverages the high-confident output of one to train the other. Besides, we introduce focal loss [17] in our method to impose more weight on ROIs of high-confidence and improve stability by removing the threshold selection. Second, ROIs of low-confidence that are ignored in the first part are used to calculate the foreground/background discrepancy between RPN and RPC. To improve the detector's transferability, the backbone network is trained to minimize the discrepancy while RPN and RPC try to maximize it. We verify its effectiveness under different adaptation scenarios.

To sum up, this work has the following contributions: (1) We are the first to reveal the significance of exploring the transferability of RPN module for domain-

[1] Noted that the recall computation of RPC here refers to the proportion of detected ROIs (not GT objects) having correct label prediction.

adaptive object detection. Simple alignment in backbone can not guarantee that the RPC receives high quality proposals. (2) From the perspective of treating RPN and RPC independently, we derive a collaborative self-training method that can propagate the loss gradient through the whole network and mutually enhance each other. (3) To the best of our knowledge, we are the first to adapt Maximum Classifier Discrepancy, MCD [26] to two-stage object detection framework for domain adaptation by focusing on ambiguous ROIs and show its effectiveness.

## 2 Related Works

***Object Detection.*** Object detection has been around for a long time and is now an important research topic in computer vision. The development of convolutional neural networks has greatly advanced the performance of object detection. CNN-based detectors can be mainly divided into two categories: one-stage detectors and two-stage detectors. Although one-stage detectors such as YOLO [22] and SSD [18] have notably higher efficiency and have become popular paradigms, two-stage detectors like Fast RCNN [9], Faster RCNN [23] and Mask RCNN [11] are still widely adopted for their much higher performance. Faster RCNN [23] is a classical two-stage object detector and is commonly used as a baseline for domain adaptation. However, object detectors suffer from domain gap when being applied to an unseen domain. Generally the backbone network pre-trained with ImageNet [5] is fine-tuned on large amount of object-level labeled data for detection together with RPN and RPC. Unfortunately, such annotated data is usually prohibitive in target domains.

***Domain Adaptation.*** Domain adaptation aims to utilize the labeled source domain data and unlabeled target domain data to boost performance on the latter. Domain adaptation on classification has been widely studied with technical paradigms like subspace alignment [6], asymmetric metric learning [16] and covariance matrix alignment [29]. A typical approach for domain adaptation is to reduce domain gap by making features or images from the two domains indistinguishable. Some methods try to minimize the distance between features of the two domains by resorting to MMD [1] or $\mathcal{H}\Delta\mathcal{H}$ [26], while some of the other works employ adversarial training with gradient reverse layer [7] or use generative adversarial networks [2,33]. Besides, entropy minimization has also been applied to domain adaptation for classification [19] and segmentation [31]. Domain-adaptive object detection has a completely different framework from image classification and semantic segmentation. It includes both object proposal detection and region-level classification. Therefore, when designing a domain-adaptive detection algorithm, it is far from sufficient to simply consider the alignment of the backbone network features, and it is necessary to consider the transferability of the algorithm in both RPN and RPC.

***Domain Adaptation for Object Detection.*** In the past few years there has been some research in domain adaptation for object detection [12,15,24, 30,32]. Raj *et al.* [21] first proposed a domain adaptation method on RCNN

with subspace alignment. Chen *et al.* [3] used a global-level and an instance-level alignment method respectively for the global and regional features. Inoue *et al.* [13] proposed a weakly-supervised framework with pseudo label. Saito *et al.* [25] pointed out that global alignment on complicated scenes and layouts might lead to negative transfer, and they proposed to employ a weak global alignment in the final layer of the backbone network, which puts more emphasis on images that are globally similar, and a strong local alignment in lower layer of the backbone. Zhu *et al.* [34] utilized RPN proposals to mine the discriminative regions of fixed size, which are pertinent to object detection, and focused on the alignment of those regions. These methods mainly focus on the alignment of the backbone stream regardless of the RPN transferability.

## 3 Method

### 3.1 Framework Overview

Given one labeled dataset from the source domain and an unlabeled one from the target domain, our task is to train a detector to obtain the best performance on the target dataset. For simplicity, both datasets share the same label space. Traditionally, feature-level domain adaptation methods try to extract the domain-invariant feature from both datasets, neglecting the adaptation of the main modules (i.e., RPN and RPC) besides the backbone stream.

The architecture of our model is illustrated in Fig. 3. The blue region on the top includes the modules in Faster RCNN, in which RPN generates and sends ROIs to the head of RPC for ROI-pooling. The yellow part is a domain discriminator that tries to determine which domain the input features originate from. It takes the backbone feature as input and outputs a domain prediction map of the same size. Besides, RPN prediction is used to highlight the foreground anchors in discriminator loss calculation. The red part consists of the proposed collaborative self-training scheme and the discrepancy maximization/minimization between RPN and RPC. ROIs with high-confidence of RPN (RPC) are used to train RPC (RPN), while ambiguous ROIs are used for discrepancy optimization. We illustrate the mutual complementary relationship of the proposed collaborative training and MCD optimization in Fig. 2. Among them, for collaborative training, the higher the confidence level of the ROI, the greater the weight will be given when calculating the loss function, while the opposite is true for MCD optimization. The lower the confidence level of ROI, the larger the sample weight will be. The two curves are implemented by tailor-designed polynomial functions in our experiment.

### 3.2 Collaborative Self-training

Generally a prevailing two-stage detector can be separated into three parts: the backbone F, RPN and RPC. Backbone F plays the role of feature representation and extracts the feature of the whole image. Then RPN takes the feature as

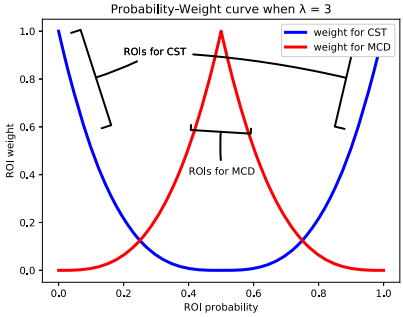



**Fig. 2.** The weight of ROI w.r.t its probability in loss calculation. Blue curve is used for collaborative self-training, and the red curve is for MCD. (Color figure online)

input, and predicts the foreground/background score of each anchor across the feature map. ROI pooling is applied to the anchors of high foreground probability for feature extraction and further sent to the RPC. Finally, RPC performs category prediction and regression of size and position of bounding boxes.

Although RPC performs regional classification based on the resulted proposals of the RPN module, it does not back propagate gradient to RPN during training. RPN filters out anchors with low foreground probability before feeding the rest to the RPC module. If the proposal filtering operation is removed and the RPN module performs ROI pooling at each anchor, the RPC module can be considered equivalent to the RPN. Ideally, the outputs of RPN and RPC should also be consistent. Those anchors with high background score in RPC should have low RPN foreground probability. Similarly, anchors having high score with non-background classes should also have high RPN foreground probability.

The core motivation of this paper is to improve the performance of object detection in the target domain by fully exploiting the domain-adaptive capability of the RPN module. Now we have accessed to the labeled image $x_s$ and its annotation $y_s$ from an annotated source dataset $\{X_s, Y_s\}$, as well as the unlabeled image $x_t$ drawn from an unlabeled target dataset $\{X_t\}$. For labeled source image $x_s$, we introduce supervised training loss of Faster RCNN, which is calculated as follows [3]:

$$L_{det} = L_{rpn}(x_s, y_s) + L_{cls}(x_s, y_s). \tag{1}$$

As we do not have accessed to the annotations $\{y_t\}$, we mutually train the two modules of RPN and RPC by leveraging the output of one module to train the other. Given a target image $x_t$, feature $f_t$ is first extracted by the feature extractor $F$. Based on $f_t$, RPN predicts the score (i.e., foreground and background probability) $s_{rpn}$ for each ROI $r_t$ while RPC outputs its class probability distribution $s_{cls}$, including the score of the background category and several foreground ones. For those ROIs with high-confident $s_{cls}$, we reuse them to update RPN, the loss of which is calculated as:

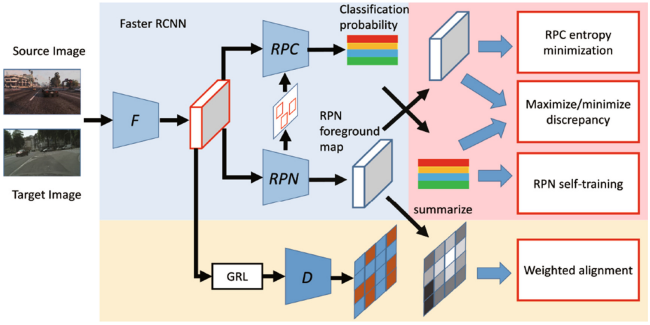



**Fig. 3.** Network architecture of our method. GRL stands for gradient reverse layer. From top to bottom: entropy minimization on RPC probability weighted by corresponding foreground score in RPN foreground map; discrepancy calculation on ambiguous ROIs shared in both RPN and RPC; RPN self-training using high-confident RPC detection results; domain discriminator loss weighted by summarized foreground probability in the same position.

$$L_{rpn_t} = f_w(s_{cls}) L_{rpn}(x_t, \hat{y}_t), \tag{2}$$

where $f_w$ can be defined as any function that decreases when $s_{cls}$ is uncertain on the foreground/background classification. For simplicity, we define it as:

$$f_w(s_{cls}) = (|1 - 2s_{cls}^{bg}|)^{\lambda}, \tag{3}$$

where $s_{cls}^{bg}$ is the background score in $s_{cls}$. $\lambda$ controls the weight on samples of low-confidence. Besides, $\hat{y}_t$ in Eq. 2 refers to the pseudo label which contains ROIs with high-confident $s_{cls}$, including both foreground and background region proposals. It does not need to contain every object in the original image nor the specific class. In Faster RCNN, RPN is trained in a selective way. Assuming that the feature extractor forwards a feature map of spatial size $H \times W$, there will generally be $H \times W \times 9$ anchors and a prediction map of equal size. Only a small portion of anchors are referenced in $L_{rpn}$ calculation and missing labels do not hurt the performance of RPN without extra processing.

On the other side, we can perform similar operations in the RPC module. Since RPN focuses on foreground/background classification and can not provide anchors with category-level pseudo labels which is necessary for RPC training, we adopt entropy minimization [19] for RPC, and adaptively assign higher weight to the samples of high-confidence in the calculation of the loss function. Based on entropy minimization, RPC is trained to output high-confident $s_{cls}$ for ROIs with high-confident $s_{rpn}$. Similar to the RPN module, we define:

$$L_{cls_t} = f_w(s_{rpn}) E(s_{cls}), \tag{4}$$

$$E(s_{cls}) = -\sum_{c \in C} s_{cls}^{c} log(s_{cls}^{c}), \tag{5}$$

$$f_w(s_{rpn}) = |1 - 2s_{rpn}^{fg}|^{\lambda}, \tag{6}$$

where $C$ includes the background and all foreground classes. $s_{cls}^{c}$ denotes the predicted probability of class $c$ in $s_{cls}$ while $s_{rpn}^{fg}$ refers to the output foreground probability of the RPN module.

### 3.3 Maximize Discrepancy Classifier on Detectors

As described above, the loss term calculation of each ROI is multiplied by an adaptive weight during the collaborative self-training process. As shown in Fig. 2, the weight value of RPN update depends on the relevant output score of the RPC branch, and vice versa. We design the weight function and guide the training process to focus more on ROIs with high-confidence. In this section, we creatively introduce a customized maximizing discrepancy classifier, i.e., MCD [26], and point out that those ROIs with low-confidence can also be effectively leveraged to improve the model adaptation. MCD is a method originally proposed for domain adaptive image classification, which utilizes task-specific classifiers to align the distributions of source and target. It works by first separating the network into the feature extractor and classifier, and duplicating the latter. During training, the two classifiers learn to maximize the prediction discrepancy between themselves while the feature extractor tries to minimize it. MCD theoretically pointed out that by minimizing and maximizing discrepancy, the transferability of the model can be effectively improved. We borrow it here and formulate the two-stage classification process in detection to satisfy its setting, which further complements the collaborative self-training. Specifically, we regard RPN and RPC as two foreground/background classifiers without duplication but weight ROIs in an opposite way. As the red curve shown in Fig. 2, we assign higher weight to ROIs with low-confidence when performing MCD loss calculation. The discrepancy between RPN and RPC is defined as:

$$s_{cls}^{fg} = \sum_{c \in C} s_{cls}^{c}, \tag{7}$$

$$L_{discrepancy}(s_{cls}, s_{rpn}) = |s_{cls}^{fg} - s_{rpn}^{fg}|, \tag{8}$$

where $C$ is the set of foreground categories. $L_{discrepancy}$ measures the foreground/background discrepancy between RPN and RPC based on their predictions. In addition, we define the weight function as:

$$f_w(s_{cls}, s_{rpn}) = (2min(|s_{cls}^{fg}|, |1 - s_{cls}^{fg}|, |s_{rpn}^{fg}|, |1 - s_{rpn}^{fg}|))^{\lambda}. \tag{9}$$

$f_w(s_{cls}, s_{rpn})$ is set to obtain a larger value when both $s_{cls}^{fg}$ and $s_{rpn}^{fg}$ are around 0.5 (i.e., of low-confident prediction), and a smaller value when either of them approaches 0 or 1. It is introduced here to mitigate the negative impact of noisy RPN prediction in the calculation of MCD loss, which is defined as,

$$L_{MCD} = f_w(s_{cls}, s_{rpn}) L_{discrepancy}(s_{cls}, s_{rpn}). \tag{10}$$

The feature extractor $F$ is trained to minimize the $L_{MCD}$ while RPN and RPC try to maximize it alternately. As shown in Fig. 4, each curve represents a decision boundary of a specific classifier. In this case, they are replaced by RPN and RPC. Samples between two decision boundaries are more likely to be wrongly classified while those far from decision boundaries are more similar to the source domain and thus the output of which can be regarded as reliable pseudo labels.

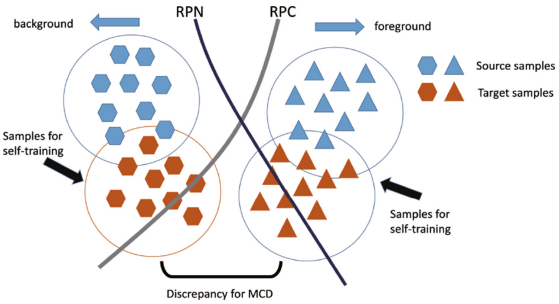



**Fig. 4.** Illustration of MCD principle. Samples far from the two decision boundaries of RPN and RPC tend to be reliable while others between the two boundaries are utilized for discrepancy computation.

### 3.4 RPN Weighted Alignment

Feature alignment between domains is a basic strategy often used by domain adaptive algorithms, and its effectiveness is widely recognized. As shown in Fig. 3, we further introduce a domain discriminator to achieve cross-domain feature alignment, and verify that it stabilize and is complementary to the previously introduced collaborative self-training paradigm. However, due to the diversity of object category combinations and the scene complexity in object detection, simply aligning the whole image might lead to failure. Some works have attempted to solve this problem, for example Saito *et al.* [25] use a weak global alignment and Zhu *et al.* [34] try to align the ROIs after region mining. We instead design a fine-grained and more flexible alignment with RPN score, i.e., the discriminator focusing on the regions of higher foreground probability.

Specifically, we design RPN weighted alignment as a local domain discriminator [25]. Discriminator $D$ takes the feature of size $H \times W \times C$ from the backbone as input, and outputs a $H \times W$ probability map, the value of which denotes the domain probability of the specific position. We scale the RPN foreground map $f$ to the same spatial size $(H \times W \times 9)$ and weight the loss as follows:

$$L_{adv_s} = \frac{1}{HW} \sum_{w=1}^{W} \sum_{h=1}^{H} (D(F(x_s))_{wh}^2 \sum_{i=1}^{9} (f_i)_{wh}), \tag{11}$$

$$L_{adv_t} = \frac{1}{HW} \sum_{w=1}^{W} \sum_{h=1}^{H} ((1 - D(F(x_t))_{wh})^2 \sum_{i=1}^{9} (f_i)_{wh}). \tag{12}$$

$f_i$ is the i-th channel of the RPN foreground map $f$ with size $H \times W$, which represents the probability that the i-th anchor box defined at each position belongs to the foreground. $(f_i)_{wh}$ and $D(F(x))_{wh}$ are the element of $f_i$ and $D(F(x))$ at position $(w, h)$. The foreground map $f$ might be rough at the beginning, but it will continue to be optimized as the collaborative self-training iterates and become an effective complement to the collaborative self-training at the backbone.

### 3.5 Overall Objective

The detection loss of original Faster RCNN consists of localization loss $L_{rpn}$ calculated on RPN and classification loss $L_{cls}$ on RPC. For source image, loss is defined as follows:

$$L_s = L_{rpn} + L_{cls} + L_{adv_s}. \tag{13}$$

For the target domain, it is slightly different due to $L_{MCD}$. we define the backbone loss and the loss of RPN and RPC as:

$$L_{t_{backbone}} = L_{adv_t} + \alpha L_{rpn_t} + \beta L_{cls_t} + \gamma L_{MCD}, \tag{14}$$

$$L_{t_{RPN,RPC}} = \alpha L_{rpn_t} + \beta L_{cls_t} - \gamma L_{MCD}, \tag{15}$$

and the loss for the discriminator is:

$$L_D = L_{adv_s} + L_{adv_t}. \tag{16}$$

$\alpha, \beta, \gamma$ control the trade-off between the detection loss of the source image and other losses. As [7], we adopt GRL (gradient reverse layer), which flips the sign of gradients in back-propagation, to implement the adversarial loss.

## 4 Experiment

We adopt unsupervised domain adaptation protocol and use four datasets in our experiments for evaluation, including Cityscapes [4], FoggyCityscapes [27], Sim10k [14] and KITTI [8]. Both images and annotations are provided for the source domain, while only images are available for target domain at training. As with [3] and [34], we evaluate our method on three kinds of domain shifts.

### 4.1 Implement Details

Following [3,25,34], we adopt Faster RCNN [23] with VGG16 [28] backbone and ROI-alignment [11] in all our experiments. We resize the input images so that the length of the shorter size is 600 and keep the aspect ratio following [25]. Our model is trained with three steps using SGD with 0.9 momentum and 0.001 learning rate. We first pre-train the model using the source dataset, followed by 10,000 iterations with $L_{det}$ calculated on the source domain and $L_{adv}$ on both domains, and finally train the network with all loss terms in Sect. 3.5 for 6,000 iterations. Without specific notation, we set $\alpha$ as 0.1, $\beta$ as 0.05, $\gamma$ as 0.1. For simplicity, pseudo boxes with confidence under 0.9 are discarded in RPN self-training and $f_w(s_{cls})$ is set to 1. $\lambda$ is set to 2 in $L_{MCD}$ and 5 in other loss terms. We implement all methods with Pytorch [20]. The architecture of domain discriminator follows [25].

We compare our method with four baselines: Faster RCNN [23], domain adaptive Faster RCNN (DA-Faster) [3], Strong Weak alignment (SWDA) [25], and selective cross-domain alignment (SCDA) [34]. The Faster RCNN model is only trained with the source images and labels without referring to the target data, which is also referred to as the source only model.

### 4.2 Domain Adaptation for Detection

***Normal to Foggy.*** Cityscapes [4] is used as the source domain while FoggyCityscapes [27] as the target. In both domains, we use the training set for pre-training and adaptation without augmentation, and evaluate our model on the validation set for 8 classes. The results are reported in Table 1. As shown in the table, our method outperforms the existing state-of-the-art by 1.6% w.r.t mAP. Besides, our method outperforms existing methods in class *car*, which is the most common object in target domain.

***Synthetic to Real.*** Sim10k [14] is used as the source domain and Cityscapes as the target domain. Similar to [3,25,34], we evaluate the detection performance on *car* in Cityscapes validation set. The results of our method is reported in Table 2. Our method outperforms the existing state-of-the-art method by 1.5% w.r.t mAP.

***Cross Camera Adaptation.*** We used KITTI [8] as source domain and Cityscapes as the target domain for evaluation. The results under this scenario is reported in Table 3. Our method improves the existing best method by 1.1% w.r.t mAP.

**Table 1.** AP(%) from Cityscapes to FoggyCityscapes

| Method | Person | Rider | Car | Truck | Bus | Train | Motobike | Bicycle | mAP |
|---|---|---|---|---|---|---|---|---|---|
| Faster RCNN [23] | 29.7 | 32.2 | 44.6 | 16.2 | 27.0 | 9.1 | 20.7 | 29.7 | 26.2 |
| DA-Faster [3] | 25.0 | 31.0 | 40.5 | 22.1 | 35.3 | 20.2 | 20.0 | 27.1 | 27.6 |
| SWDA [25] | 29.9 | 42.3 | 43.5 | 24.5 | 36.2 | **32.6** | 30.0 | 35.3 | 34.3 |
| SCDA [34] | **33.5** | 38 | 48.5 | **26.5** | 39 | 23.3 | 28 | 33.6 | 33.8 |
| Proposed | 32.7 | **44.4** | **50.1** | 21.7 | **45.6** | 25.4 | **30.1** | **36.8** | **35.9** |

**Table 2.** AP on "Car"(%) from Sim10k to Cityscapes.

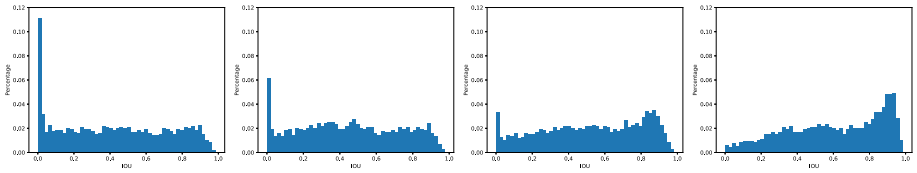
| Method | AP on "Car" |
|---|---|
| Faster RCNN [23] | 34.57 |
| DA-Faster [3] | 38.97 |
| SWDA [25] | 40.10 |
| SCDA [34] | 43.05 |
| Proposed | 44.51 |

**Table 3.** AP on "Car"(%) from KITTI to Cityscapes.

| Method | AP on "Car" |
|---|---|
| Faster RCNN [23] | 34.9 |
| DA-Faster [3] | 38.5 |
| SWDA [25] | – |
| SCDA [34] | 42.5 |
| Proposed | 43.6 |

## 5 Ablation Study

***Effectiveness of RPN Adaptation.*** As one of our core motivations is to explore the significance of the RPN module for domain adaptation. We verify the superiority of collaborative self-training by analyzing the quality of region proposals generated by different adaptation models. We first define a metric called *proposal coverage*. Given a ground truth bounding box, we define the largest IOU with all detected proposals as its proposal coverage. For each ground truth in the target domain, we calculate the proposal coverage and count the distribution for each detection model. We conduct experiments on the adaptation from Sim10k to the Cityscapes dataset for comparison. Firstly, in order to verify that the naive local alignment in the backbone branch is not sufficient for domain adaptive object detection, we adopt a non-weighted local alignment method as naive alignment for comparison, in which a domain discriminator is applied to every position of the feature map. It can be implemented by removing the $f_i$ weighting strategy in Eq. 11 and Eq. 12. We compare the proposal coverage distribution of four different detection models, including the source-only model, naive alignment model, our RPN adaptation with collaborative self-training and the Oracle model (i.e., the performance upper bound which refers to the annotations of the target domain). As shown in Fig. 5, our method greatly improves the quality of the generated object proposals in the target domain and its proposal coverage distribution is much more closer to the Oracle model, when compared with the naive alignment. Specifically, our method greatly reduces the boxes with proposal coverage = 0 compared with both the



(a) Source Only (b) Naive alignment (c) RPN adaption (d) Oracle

**Fig. 5.** Proposal coverage distribution of different proposal generation models. "Naive alignment" stands for non-weighted local alignment and "Oracle" refers to the model trained with labeled target dataset. RPN adaption is trained using our proposed collaborative self-training paradigm.

source only and naive alignment models. It also obviously improves the quality of proposals with IOU $\geq 0.5$ while the naive alignment mainly changes the distribution of IOU $< 0.5$ w.r.t the source-only baseline. This shows that our method can effectively improve the accuracy of the generated proposals, and therefore bring about significant numerical performance improvements (Table 4). Noted that the performance benefit may become more apparent as the IOU threshold setting to a higher value.

***Effectiveness of Different Components.*** We evaluate the contribution of different components by designing several variants of our model. The results are reported in Table 4. All experiments are conducted on the adaptation from Sim10k to Cityscapes dataset. Hyper-parameter setting of all model variants remain the same as described in Sect. 4.1. As shown in the table, incorporating the core module, i.e., collaborative self-training (CST), to the baseline source only model can bring significant performance gain of 7.76% w.r.t AP, increasing AP from 34.57% to 42.33%. This reveals that the domain adaptation capability can indeed be improved by effectively mining the complementary advantages of RPN and RPC. On the other hand, applying a naive local alignment only results in 2.46% performance improvement, which proves that simple alignment on the backbone branch is far from sufficient for domain adaptive object detection. Nevertheless, the proposed weighted local alignment still outperforms the naive alignment by 1.28% w.r.t AP even without RPN self-training. It is worth noting that using MCD alone does not significantly improve the baseline (36.42% VS 34.57%) because most of the uncertain ROIs are filtered out by RPN. Last but not least, as can be seen from the last two rows of the table, CST is complementary to the RPN weighted alignment and MCD, our entire model gains an additional 2.18% AP when compared to the CST-only version. In general, all three proposed components make their own contributions compared with the source-only model, which overall improve the baseline by 9.94% w.r.t AP.

***Visualization of Detection Results.*** The detection results after adaptation are illustrated in Fig. 6. Our model can accurately localize and classify objects under different kinds of domain shifts. We also visualize the weight used in the local alignment calculation. It is obvious that RPN weighted method can

**Fig. 6.** Detection result on target domain. From left to right: Sim10k to Cityscapes; Kitti to Cityscapes; Cityscapes to FoggyCityscapes. The second row shows the weight inferred from our RPN weighted local domain discriminator. Brighter colors indicate higher attention. Apparently, regions with considered objects (e.g. car) are of higher weight in loss calculation. (Color figure online)

**Table 4.** AP on "Car" from Sim10k to Cityscapes of different method variants to demonstrate the effectiveness of the proposed algorithm. "Local" represents naive local alignment and "CST" refers to the collaborative self-training.

| Local | Weight local | CST | MCD | AP on Car |
|---|---|---|---|---|
| | | | | 34.57 |
| ✓ | | | | 37.03 |
| | ✓ | | | 38.31 |
| | | ✓ | | 42.33 |
| | | | ✓ | 36.42 |
| | ✓ | ✓ | | 43.08 |
| | ✓ | ✓ | ✓ | 44.51 |

effectively suppress non-critical parts of the image. As shown in the Figure, although the sky and roads occupy most of the area of the image, the inferred weight map shows that these areas have little effect on distinguishing objects of different domains, which is consistent with our optimization goal.

## 6 Conclusion

In this paper, we are the first to empirically reveal that the RPN and RPC module in the endemic two-stage detectors (e.g., Faster RCNN) demonstrate significantly different transferability when facing large domain gap. Base on this observation, we design a collaborative self-training method for RPN and RPC to train each other with ROIs of high-confidence. Moreover, a customized maximizing discrepancy classifier is introduced to effectively leverage ROIs with low-confidence to further increase the accuracy and generalization of the detection model. Experimental results demonstrated that our method significantly

improves the transferability and outperforms existing methods in various domain adaptation scenarios.

**Acknowledgements.** This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation (2020B1515020048), in part by the National Natural Science Foundation of China (61976250, 61702565, U1811463), in part by the National High Level Talents Special Support Plan (Ten Thousand Talents Program), in part by the Fundamental Research Funds for the Central Universities (18lgpy63). This work was also sponsored by Meituan-Dianping Group.

## References

1. Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., Smola, A.J.: Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics **22**(14), e49–e57 (2006)
2. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3722–3731 (2017)
3. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster R-CNN for object detection in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3339–3348 (2018)
4. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
6. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2960–2967 (2013)
7. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(1), 2096–2130 (2016)
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361. IEEE (2012)
9. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
10. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
12. Hoffman, J., et al.: CyCADA: cycle-consistent adversarial domain adaptation. arXiv preprint arXiv:1711.03213 (2017)
13. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5001–5009 (2018)

14. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R.: Driving in the matrix: can virtual worlds replace human-generated annotations for real world tasks? arXiv preprint arXiv:1610.01983 (2016)
15. Kim, T., Jeong, M., Kim, S., Choi, S., Kim, C.: Diversify and match: a domain adaptive representation learning paradigm for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12456–12465 (2019)
16. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In: CVPR 2011, pp. 1785–1792. IEEE (2011)
17. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
18. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
19. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: Advances in Neural Information Processing Systems,. pp. 136–144 (2016)
20. Paszke, A., et al.: Automatic differentiation in PyTorch (2017)
21. Raj, A., Namboodiri, V.P., Tuytelaars, T.: Subspace alignment based domain adaptation for RCNN detector. arXiv preprint arXiv:1507.05578 (2015)
22. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
24. RoyChowdhury, A., et al.: Automatic adaptation of object detectors to new domains using self-training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 780–790 (2019)
25. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. arXiv (2018)
26. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723–3732 (2018)
27. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. Int. J. Comput. Vis. **126**(9), 973–992 (2018)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
29. Sun, B., Feng, J., Saenko, K.: Return of frustratingly easy domain adaptation. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)
30. Tzeng, E., Burns, K., Saenko, K., Darrell, T.: SPLAT: semantic pixel-level adaptation transforms for detection. arXiv preprint arXiv:1812.00929 (2018)
31. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: ADVENT: adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2517–2526 (2019)

32. Wang, T., Zhang, X., Yuan, L., Feng, J.: Few-shot adaptive faster R-CNN. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7173–7182 (2019)
33. Yoo, D., Kim, N., Park, S., Paek, A.S., Kweon, I.S.: Pixel-level domain transfer. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 517–532. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_31
34. Zhu, X., Pang, J., Yang, C., Shi, J., Lin, D.: Adapting object detectors via selective cross-domain alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 687–696 (2019)